Bilkent Üniversitesi

Database Systems

Department of Computer Engineering
Spring 2021

Project Proposal

Online Course Platform - Sapientia

Project Group 4:

- Ömer Ünlüsoy - Section 01 - 21702136
- Elif Gamze Güliter - Section 01 - 21802870
- İrem Tekin - Section 01 - 21803267
- Ece Ünal - Section 01 - 21703149

Supervisor:

- Uğur Güdükbay

Introduction

In this project, we are going to implement a web-based course platform similar to Udemy [1]. The purpose of the platform is to enable instructors to build their online courses as well as students to reach out these contents and educate themselves. The name of the platform is **Sapientia**.

In Sapientia, instructors will upload courses about the topics that they are proficient about and students will purchase these courses in order to enroll. A course will consist of several lectures as well as quizzes and students will receive a certificate indicating that they have finished the online course if only if they watch all of the lectures and take all the quizzes. In addition, Sapientia consists of students' wishlist, students' notes for a specific lecture, Q&A section for students and instructors to interact with each other, ratings of courses, discount or refund for courses, and at last, announcements made by instructors about a course.

The report continues with a Description section where, how and why the database will be part of the online course platform are explained. Following that, in the Requirements section we will give information about the functional and non-functional requirements and the constraints of our project. Then, we will indicate the limitations of the platform under the section Limitations. At last, the report will be finished with the Conceptual Design section where the E-R Diagram of our project will be given.

Description

Sapientia lets people share their skills with masses. This web based online course application proves that everyone can be an instructor or a student. Instructors can share quizzes relevant to the subject that they teach, to observe what are the most challenging subjects to students, and can focus on them. Also, with the help of quizzes, students can be aware of their weak points about the subject. If they are stuck in some part and can't figure out with the help of what they learnt in the

lectures, they can ask their questions directly to the course instructor in the Q&A section. Additionally, instructors can arrange live problem sessions via Zoom links. Students can get a certificate as a proof that they finished the course. Additionally, students can add courses they plan to take to their wishlist. During lectures, they can take notes to review them afterwards. Site admin can apply a discount to the course price if the instructor allows this option. Students can send their complaints about the course to get a refund to the site admin. If the site admin thinks that the reason is valid for getting a refund, students can take their money back and drop the course. Finally, instructors can request statistics and help from the site admin.

a. Why database is going to be a part of the system

Online course platform web application, will store different types of information such as courses, lectures, quizzes, question forums, complaints, announcement, certificate, wishlist, user types and their data, etc. In order to store, update, and retrieve data, a database is necessary. Furthermore, users of this platform (students, instructors and admin) will have different access rights and their functions in the platform will vary which make the platform require an efficient database management system. Using a database in this project enables us to store and manage large amounts of data, secure the data with different access rights, retrieve data easily and ensure accurate data using constraints.

b. How database is going to be a part of the system

The database system will be used to store, manage and retrieve data to achieve a dynamic, secure, functioning online course platform. Users of the platform will be given access rights to data according to their types, information about courses, lectures, payments, Q&A sections, users will be stored, that information will be updated and retrieved by queries using the database. Grouping the data according to some criteria, relations between different data groups will also be managed by the database system.

Requirements

a. Functional Requirements

Functional Requirements is separated into three different categories depending on the user type; students, instructors (course creators), and admin (site admin). Each user type has to go through an authentication step to access their accounts.

Student

- search for courses.
- add a course to wishlist.
- enroll the course they want with valid purchase info.
- take lectures of the courses they already enrolled.
- take quizzes of the courses they already enrolled if the Instructor provides.
- join active live problem sessions via Zoom.
- track their progress on the courses that they already enrolled.
- rate the course they already enrolled after finishing a certain proportion.
- request refund for the courses they enrolled with a valid reason.
- take a certificate after finishing a certain proportion.
- create notes for lectures of the courses they enrolled.
- ask questions at the Q&A section of the courses they enrolled.

Instructor

- publish their courses.
- get paid according to their course income.
- invite another Instructor to their courses.
- make announcements about their courses.
- add new lectures to their courses.
- add Quizzes to their courses.
- start live problem sessions via Zoom.
- edit the lectures of their courses.
- view the students enrolled in their courses.

- view the statistics of their courses (enrolment number, certificate number, averages).
- answer the questions at the Q&A section of their courses.

Admin

- offer discounts on courses whose creators allow.
- view complaints about courses.
- response to the complaints.
- approve or reject Students' refund requests.
- view the statistics of the courses.
- help Instructors about technical problems of their courses.

b. Non-functional Requirements

Reliability

- The system should not fail especially during the authentication of users, uploading a lecture or during signing in.

Performance

- The database design should be planned to achieve query efficiency.

Usability

- User Interface of The Website should be minimal and less challenging for non-programmer users.
- There must be an available documentation of the Website.
- The layout of the Website should be 16:9 aspect ratio.

Concurrency

- The database must be updated at each operation to keep data up to date.
- The system maintains atomicity during each operation.
- Cascading abort and cascading removal must be handled with care.

Availability

- The system should be available 7/24 in order to give uninterrupted service to the users and should be maintained regularly. If needed due to maintenance, user access to the website can be blocked but these will be announced.

Security

- The users should authenticate throughout username and password.
- Each type of the users can login to different levels, therefore, the permissions should be specified.
- There should be password specifications such as the usage of characters and numbers as well as the password length.
- The data of the users (Instructors and Students) should not be reached by the third parties.
- The data of the users should not be lost.

Privacy

- Students can keep their personal data private from the Instructors.
- Admins must not view Students’ personal data.
- Admins must not view Instructors' personal data.
- The Website must not access Students’ and Instructors’ payment information.
- Live problem sessions are held via Zoom and its privacy is not the Website’s concern.

c. Constraints

- We will use MySQL as a query language.
- For the website we will use JavaScript, HTML, CSS.
- For the backend we will use Spring framework of Java.

Limitations

- One instructor can make one announcement with the same subject name in a day about one course.
- There cannot be multiple lectures with the same title.
- Lecture duration cannot exceed 120 minutes.
- Course announcement text cannot exceed 500 characters.
- Instructors get 100% of the course income.
- Students can only ask questions and instructors can answer those questions in the Q&A section.
- A student cannot enroll in the same course more than once.
- One student cannot ask more than one question with the same subject name about one course.
- One student/instructor cannot make more than one complaint with the same subject name in a day about one course.
- A student can add a course to the wishlist once.
- A student can receive at most one certificate from a course.
- A student can be a bronze, silver or golden member if their completed course numbers are 5, 10, 20 respectively.
- A student cannot rate a course without enrollment.
- An account can only be a student account or an instructor or admin account.
- Ratings can be done out of 5. Rating of a course is rounded off to one decimal place.

Conceptual Design

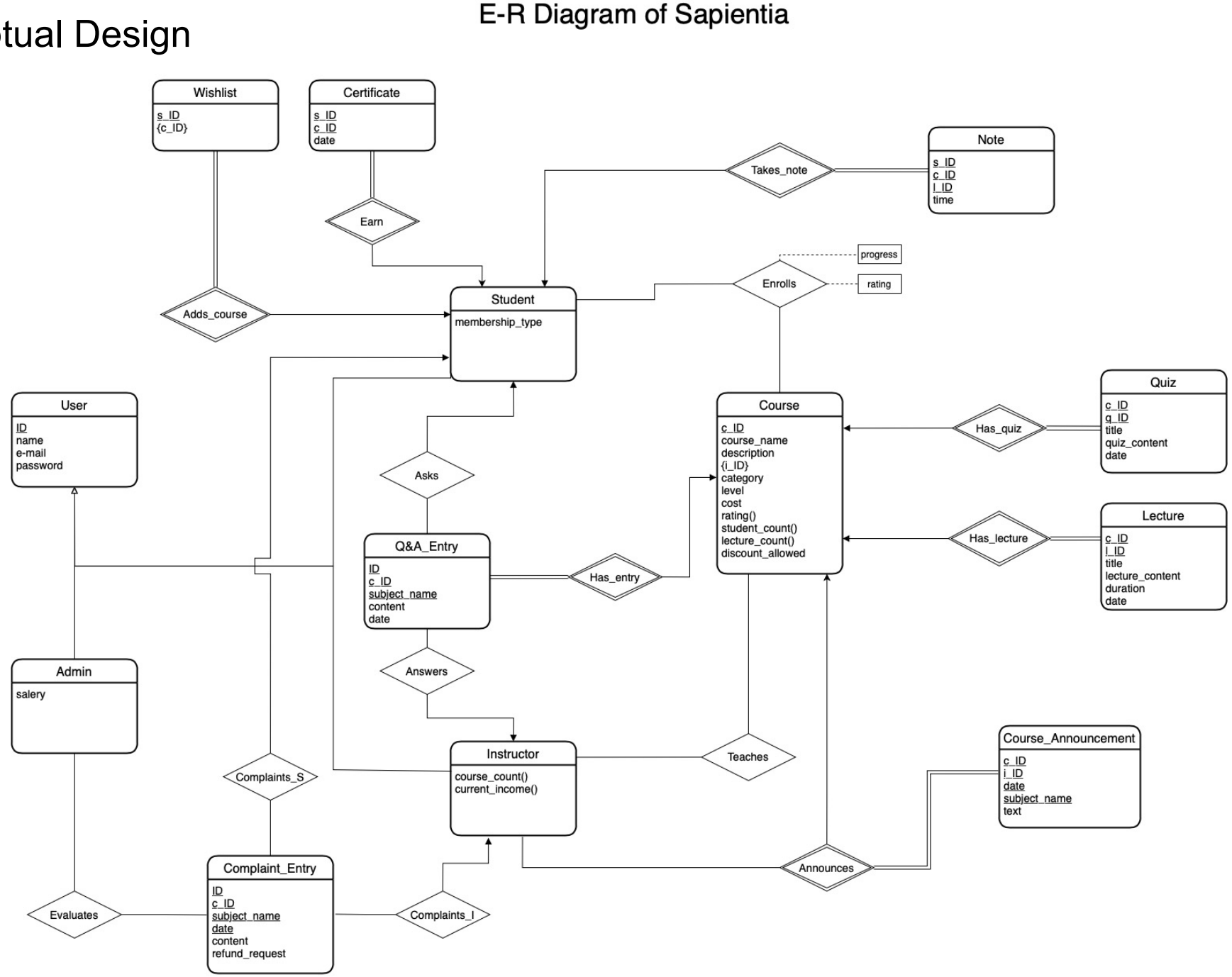

Figure 1: Entity Relationship Diagram [2]

Web Page

The link of GitHub page of the project is the following:
https://github.com/omerunlusoy/CS353_Project

References

[1] “Online Courses - Learn Anything, On Your Schedule,” *Udemy*. [Online]. Available: https://www.udemy.com. [Accessed: 21-Feb-2021].

[2] “diagrams.net - free flowchart maker and diagrams online,” *Flowchart Maker & Online Diagram Software*. [Online]. Available: https://app.diagrams.net/. [Accessed: 21-Feb-2021].